


```
CCCCCCCC HH HH MM MM MM MM RRRRRRRR KK KK CCCCCCCC HH HH GGGGGGGG
CCCCCCCC HH HH MM MM MM MM RRRRRRRR KK KK CCCCCCCC HH HH GGGGGGGG
CC CC HH HH MMMM MMMM MMMM MMMM RR RR KK KK CC CC HH HH GG
CC CC HH HH MMMM MMMM MMMM MMMM RR RR KK KK CC CC HH HH GG
CC CC HH HH MM MM MM MM MM MM RR RR KK KK CC CC HH HH GG
CC CC HH HH MM MM MM MM MM MM RR RR KK KK CC CC HH HH GG
CC CC HH HH MM MM MM MM MM MM RR RR KK KK CC CC HH HH GG
CC CC HH HH MM MM MM MM MM MM RR RR KK KK CC CC HH HH GG
CCCCCCCC HH HH MM MM MM MM MM MM RR RR RR KK KK CCCCCCCC HH HH GGGGGG
CCCCCCCC HH HH MM MM MM MM MM MM RR RR RR KK KK CCCCCCCC HH HH GGGGGG
GGGGGGGG
```

```
LL LL I I I I I SSSSSSSS
LL LL I I I I I SSSSSSSS
LL LL II
LL LL II
LL LL II
LL LL II
LL LL II
LL LL II
LL LL II
LL LL II
LL LL II
LLLLLLLLLL I I I I I SSSSSSSS
LLLLLLLLLL I I I I I SSSSSSSS
```

.....


```
0001 0 %TITLE 'EDT$CHMMRKCHG - track changes in current line'
0002 0 MODULE EDT$CHMMRKCHG ( ! Track changes in current line
0003 0 IDENT = 'V04-000' ! File: CHMMRKCHG.BLI Edit: JBS2024
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1
0031 1 ++
0032 1 FACILITY: EDT -- The DEC Standard Editor
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 This module keeps track of the bounds between which the
0037 1 current line must be updated.
0038 1
0039 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0040 1
0041 1 AUTHOR: Bob Kushlis, CREATION DATE: Unknown
0042 1
0043 1 MODIFIED BY:
0044 1
0045 1 2-001 - Recode to interface to the new screen updater. JBS 13-Sep-1982
0046 1 2-002 - Version 2-001 was just a stub. This version is little better:
0047 1 it always marks the line for repainting. JBS 17-Sep-1982
0048 1 2-003 - Add logic for deleting lines and for skipping over deleted lines.
0049 1 JBS 20-Sep-1982
0050 1 2-004 - Worry about the target line having a deleted neighbor. JBS 20-Sep-1982
0051 1 2-005 - Call EDT$$SC_FNDREC, which contains the table searching code. JBS 24-Sep-1982
0052 1 2-006 - Record the actual edits to a line, instead of just marking it for repaint. JBS 28-Sep-1982
0053 1 2-007 - Change the parameters to SC_FNDREC. SMB 28-Sep-1982
0054 1 2-008 - Put in temporary zeroing of screen buffer pointer. SMB 30-Sep-1982
0055 1 2-009 - Remove yesterday's temporary change. SMB 01-Oct-1982
0056 1 2-010 - Change the way SC_FNDREC is used. SMB 08-Oct-1982
0057 1 2-011 - Don't run off the end of the edit buffer, and maintain SCR_EDIT_POS. JBS 18-Oct-1982
```

:	58	0058	1	:	2-012	- Turn insert followed by delete into a null operation. JBS 24-Oct-1982
:	59	0059	1	:	2-013	- Fix problems involving marking deleted characters. JBS 24-Oct-1982
:	60	0060	1	:	2-014	- If we free the first screen pointer, update EDT\$SA_FST_SCRPTR.
:	61	0061	1	:	2-015	- Maintain SCR_EDIT_MINPOS. JBS 28-Oct-1982
:	62	0062	1	:	2-016	- Use EDT\$SSC_ENDEL to free line blocks. JBS 29-Oct-1982
:	63	0063	1	:	2-017	- Fix a bad LOCAL declaration. SMB 31-Oct-1982
:	64	0064	1	:	2-018	- Maintain MINPOS, even if we must repaint, to avoid extra repainting
:	65	0065	1	:		if we overflow the edit buffer, which is likely on the PDP-11. JBS 01-Dec-1982
:	66	0066	1	:	2-019	- Maintain MINPOS even if the previous edit is a repaint. JBS 04-Dec-1982
:	67	0067	1	:	2-020	- If we are editing the first character of a continuation line, mark the
:	68	0068	1	:		previous line as needing repainting at its end, in case we are inserting
:	69	0069	1	:		a character before a wide character that should really be painted at the
:	70	0070	1	:		end of the previous line. JBS 20-Dec-1982
:	71	0071	1	:	2-021	- Remove the edit buffer. JBS 27-Dec-1982
:	72	0072	1	:	2-022	- If we delete an inserted continuation or continued line,
:	73	0073	1	:		rebuild the screen data structure. JBS 05-Jan-1983
:	74	0074	1	:	2-023	- If we are going to rebuild the screen data structure,
:	75	0075	1	:		don't update anything. JBS 17-Feb-1983
:	76	0076	1	:	2-024	- Set EDT\$G_ANY_CHANGES if any changes. JBS 05-Apr-1983
:	77	0077	1	:	--	
:	78	0078	1	:		

EDT\$CHMMRKCHG
V04-000

EDT\$CHMMRKCHG - track changes in current line
Declarations

E 3
16-Sep-1984 00:04:32
14-Sep-1984 12:22:39

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]CHMMRKCHG.BLI;1

Page 3
(2)

```
: 80      0079 1 %SBTTL 'Declarations'
: 81      0080 1
: 82      0081 1 | TABLE OF CONTENTS:
: 83      0082 1 |
: 84      0083 1 |
: 85      0084 1 REQUIRE 'EDT$RC:TRAROUNAM';
: 86      0523 1
: 87      0524 1 FORWARD ROUTINE
: 88      0525 1     EDT$SMRK_LNCHG : NOVALUE;
: 89      0526 1
: 90      0527 1 |
: 91      0528 1 | INCLUDE FILES:
: 92      0529 1 |
: 93      0530 1 |
: 94      0531 1 REQUIRE 'EDT$RC:EDTREQ';
: 95      0666 1
: 96      0667 1 |
: 97      0668 1 | MACROS:
: 98      0669 1 |
: 99      0670 1 |     NONE
100      0671 1 |
101      0672 1 | EQUATED SYMBOLS:
102      0673 1 |
103      0674 1 |     NONE
104      0675 1 |
105      0676 1 | OWN STORAGE:
106      0677 1 |
107      0678 1 |     NONE
108      0679 1 |
109      0680 1 | EXTERNAL REFERENCES:
110      0681 1 |
111      0682 1 |     In the routine
```

! Mark changes in the current line

**F

```
113 0683 1 %SBTTL 'EDT$SMRK_LNCHG - mark changes in the current line'
114 0684 1
115 0685 1 GLOBAL ROUTINE EDT$SMRK_LNCHG (
116 0686 1     CODE
117 0687 1     POSITION
118 0688 1     ) : NOVALUE =
119 0689 1
120 0690 1 ++
121 0691 1 FUNCTIONAL DESCRIPTION:
122 0692 1
123 0693 1     Keep track of edits done to lines on the screen, so the screen
124 0694 1     updater can repaint them.
125 0695 1
126 0696 1 FORMAL PARAMETERS:
127 0697 1
128 0698 1     CODE           The type of edit: delete line or modify
129 0699 1     POSITION        The position in the record at which the operation takes place
130 0700 1
131 0701 1 IMPLICIT INPUTS:
132 0702 1
133 0703 1     EDT$SA_SCR_BUF
134 0704 1     EDT$SA_CUR_BUF
135 0705 1     EDT$SG_SCR_REBUILD
136 0706 1
137 0707 1 IMPLICIT OUTPUTS:
138 0708 1
139 0709 1     EDT$SG_LN_CHGD
140 0710 1     EDT$SG_SCR_REBUILD
141 0711 1
142 0712 1 ROUTINE VALUE:
143 0713 1
144 0714 1     NONE
145 0715 1
146 0716 1 SIDE EFFECTS:
147 0717 1
148 0718 1     NONE
149 0719 1
150 0720 1 --
151 0721 1
152 0722 2 BEGIN
153 0723 2
154 0724 2 EXTERNAL ROUTINE
155 0725 2     EDT$SSC_FNDREC,
156 0726 2     EDT$SSC_LNDEL : NOVALUE;
157 0727 2
158 0728 2 EXTERNAL
159 0729 2     EDT$SA_SCR_BUF,
160 0730 2     EDT$SA_CUR_BUF : REF TBCB_BLOCK,
161 0731 2     EDT$SG_LN_CHGD,
162 0732 2     EDT$SG_SCR_REBUILD,
163 0733 2     EDT$SG_ANY_CHANGES;
164 0734 2
165 0735 2 LOCAL
166 0736 2     DISP,
167 0737 2     SCRPTR : REF SCREEN_LINE,
168 0738 2     RECORD_NO : LN_BLOCK,
169 0739 2     EDIT_CODE,
```

! Mark changes in the current line
! Change code
! Position in the record

! Find a record in the screen data base
! Free a record from the screen data base

! Pointer to the current text buffer control block
! The current record has been changed
! 1 = the screen data base has to be rebuilt
! 1 = an edit has been made since last screen update

! Displacement from cursor screen pointer
! Pointer to our screen line
! Record number of our screen line
! Do this to this record


```
170 0740 2 REL_POS, ! Position in this line
171 0741 2 MIN_RELPOS; ! Minimum edit position on this line
172 0742 2
173 0743 2 +
174 0744 2 Make sure the work file system knows that the current record has been
175 0745 2 changed, so it will write the record back to the work file system.
176 0746 2 -
177 0747 2 EDT$SG_LN_CHGD = 1;
178 0748 2 +
179 0749 2 Note for the screen updater that an edit has been made. If this cell is still zero
180 0750 2 when the screen updater is called, the screen updater will not search the screen
181 0751 2 data base for lines to repaint. This improves the speed of the arrow keys.
182 0752 2 -
183 0753 2 EDT$SG_ANY_CHANGES = 1;
184 0754 2 +
185 0755 2 If we are going to rebuild the screen data base anyway, don't bother to update it.
186 0756 2 -
187 0757 2
188 0758 2 IF .EDT$SG_SCR_REBUILD THEN RETURN;
189 0759 2
190 0760 2 +
191 0761 2 Point to the current line in the screen data base.
192 0762 2 -
193 0763 2 ASSERT (.POSITION LEQ 255);
194 0764 2 SCRPTR = EDT$SC_FNDREC (.POSITION, DISP);
195 0765 2 +
196 0766 2 If the line is not in the screen data base, just return.
197 0767 2 -
198 0768 2
199 0769 2 IF (.SCRPTR EQLA 0) THEN RETURN;
200 0770 2
201 0771 2 +
202 0772 2 Check for delete of an inserted line. It is very special.
203 0773 2 -
204 0774 2 EDIT_CODE = .CODE;
205 0775 2
206 0776 2 IF ((.EDIT_CODE EQL SCR_EDIT_DELLN) AND ((.SCRPTR [SCR_EDIT_FLAGS] AND SCR_EDIT_INSLN) NEQ 0))
207 0777 2 THEN
208 0778 2 BEGIN
209 0779 2 +
210 0780 2 We are deleting an inserted line. Back out the insert so that
211 0781 2 the screen updater will see nothing.
212 0782 2 -
213 0783 2
214 0784 2 LOCAL
215 0785 2 NEXT_SCPTR : REF SCREEN_LINE;
216 0786 2
217 0787 2 +
218 0788 2 If this is a continued or continuation line, rebuild the screen data base.
219 0789 2 -
220 0790 2
221 0791 2 IF (.SCRPTR [SCR_LINE_IDX] NEQ 0)
222 0792 2 THEN
223 0793 2 BEGIN
224 0794 2 EDT$SG_SCR_REBUILD = 1;
225 0795 2 RETURN;
226 0796 2 END;
```

```
227 0797 3
228 0798 3 NEXT_SCRPTR = .SCRPTR [SCR_NXT_LINE];
229 0799 3
230 0800 4 IF (.NEXT_SCRPTR NEQA 0)
231 0801 3 THEN
232 0802 4 BEGIN
233 0803 4
234 0804 5 IF (.NEXT_SCRPTR [SCR_LINE_IDX] NEQ 0)
235 0805 4 THEN
236 0806 5 BEGIN
237 0807 5 EDT$G_SCR_REBUILD = 1;
238 0808 5 RETURN;
239 0809 4 END;
240 0810 4
241 0811 3 END;
242 0812 3
243 0813 3 EDT$SC_LNDEL (.SCRPTR);
244 0814 3 RETURN;
245 0815 2 END;
246 0816 2
247 0817 2 +
248 0818 2 - Maintain the record of the minimum edit position.
249 0819 2
250 0820 2 REL_POS = .POSITION - .SCRPTR [SCR_CHR_FROM];
251 0821 2 MIN_RELPOS = MINU (.REL_POS, .SCRPTR [SCR_EDIT_MINPOS]);
252 0822 2 SCRPTR [SCR_EDIT_MINPOS] = .MIN_RELPOS;
253 0823 2 SCRPTR [SCR_EDIT_MAXPOS] = 255;
254 0824 2 +
255 0825 2 - If we are editing at the front of a continuation line, mark the previous line as needing
256 0826 2 an edit at its end. This is in case we are inserting a narrow character in front of a
257 0827 2 wide character; the narrow character may want to be displayed at the end of the previous line.
258 0828 2
259 0829 2
260 0830 3 IF ((.REL_POS EQL 0) AND (.SCRPTR [SCR_CHR_FROM] NEQ 0)) !
261 0831 2 THEN
262 0832 2 EDT$MRK_LNCHG (SCR_EDIT_MODIFY, .POSITION - 1);
263 0833 2
264 0834 2 +
265 0835 2 - Mark the edit for this line.
266 0836 2
267 0837 2 SCRPTR [SCR_EDIT_FLAGS] = .SCRPTR [SCR_EDIT_FLAGS] OR .EDIT_CODE;
268 0838 2 +
269 0839 2 - After any edit to a line, mark all subsequent lines for the
270 0840 2 same record for repainting. This means that in NOTRUNCATE mode,
271 0841 2 if you edit a line all later lines for that record are repainted.
272 0842 2 This is important even for replaces, since the width of a character
273 0843 2 may change. Note that deleting a line causes the subsequent
274 0844 2 records to be marked as deleted.
275 0845 2
276 0846 2
277 0847 2 WHILE ((.SCRPTR [SCR_CHR_TO] NEQ 255) AND (.SCRPTR [SCR_NXT_LINE] NEQA 0)) DO
278 0848 2 BEGIN
279 0849 2 SCRPTR = .SCRPTR [SCR_NXT_LINE];
280 0850 2 SCRPTR [SCR_EDIT_MINPOS] = 0;
281 0851 2 SCRPTR [SCR_EDIT_MAXPOS] = 255;
282 0852 2 SCRPTR [SCR_EDIT_FLAGS] = .SCRPTR [SCR_EDIT_FLAGS] OR .EDIT_CODE;
283 0853 2 END;
! Repaint later lines for this record
```


EDT\$CHMMRKCHG
V04-000

EDT\$CHMMRKCHG - track changes in current line
EDT\$SMRK_LNCHG - mark changes in the current

1 3
16-Sep-1984 00:04:32
14-Sep-1984 12:22:39

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]CHMMRKCHG.BLI;1

Page 7
(3)

: 284
: 285

0854 2
0855 1 END;

! of routine EDT\$SMRK_LNCHG

.TITLE EDT\$CHMMRKCHG EDT\$CHMMRKCHG - track changes in
current line

.IDENT \V04-000\

.EXTRN EDT\$SSC_FNDREC, EDT\$SSC_LNDEL
.EXTRN EDT\$SA_SCR_BUF, EDT\$SA_CUR_BUF
.EXTRN EDT\$SG_LN_CHGD, EDT\$SG_SCR_REBUILD
.EXTRN EDT\$SG_ANY_CHANGES
.EXTRN EDT\$INTER_ERR

.PSECT _EDT\$CODE, NOWRT, SHR, PIC, 2

.ENTRY EDT\$SMRK_LNCHG, Save R2, R3, R4

MOVAB EDT\$SG_SCR_REBUILD, R4

SUBL2 #12, SP

MOVL #1, EDT\$SG_LN_CHGD

MOVL #1, EDT\$SG_ANY_CHANGES

BLBC EDT\$SG_SCR_REBUILD, 1\$

RET

CMPL POSITION, #255

BLEQ 2\$

CALLS #0, EDT\$INTER_ERR

PUSHL SP

PUSHL POSITION

CALLS #2, EDT\$SSC_FNDREC

MOVL R0, SCRPTR

BEQL 8\$

MOVL CODE, EDIT_CODE

CMPL EDIT_CODE, #4

BNEQ 5\$

BBC #1, 13(SCRPTR), 5\$

TSTB 8(SCRPTR)

BNEQ 3\$

MOVL 4(SCRPTR), NEXT_SCRPTR

BEQL 4\$

TSTB 8(NEXT_SCRPTR)

BEQL 4\$

MOVL #1, EDT\$SG_SCR_REBUILD

RET

PUSHL SCRPTR

CALLS #1, EDT\$SSC_LNDEL

RET

MOVZBL 9(SCRPTR), REL_POS

SUBL3 REL_POS, POSITION, REL_POS

MOVL REL_POS, R1

CMPZV #0, #8, 11(SCRPTR), R1

BGEQU 6\$

MOVZBL 11(SCRPTR), R1

MOVB MIN_RELPOS, 11(SCRPTR)

MNEGB #1, 12(SCRPTR)

TSTL REL_POS

BNEQ 7\$

TSTB 9(SCRPTR)

: 0685

: 0747

: 0753

: 0758

: 0763

: 0764

: 0769

: 0774

: 0776

: 0791

: 0798

: 0800

: 0804

: 0807

: 0806

: 0813

: 0778

: 0820

: 0821

: 0822

: 0823

: 0830

EDT\$
V04-

7E	08	AC	0C	13	00093	BEQL	7\$		
			01	C3	00095	SUBL3	#1, POSITION, -(SP)	:	0832
			01	DD	0009A	PUSHL	#1	:	
	FF5F	CF	02	FB	0009C	CALLS	#2, EDT\$SMRK_LNCHG	:	
	0D	A2	53	88	000A1	BISB2	EDIT CODE, 13(SCRPTR)	:	0837
	FF	8F	A2	91	000A5	CMPB	10(SCRPTR), #255	:	0847
			11	13	000AA	BEQL	8\$	:	
			04	A2	D5	TSTL	4(SCRPTR)	:	
			0C	13	000AF	BEQL	8\$	:	
	52	04	A2	D0	000B1	MOVL	4(SCRPTR), SCRPTR	:	0849
0B	A2	FF00	8F	B0	000B5	MOVW	#65280, 11(SCRPTR)	:	0850
			E4	11	000BB	BRB	7\$	:	0852
			04	000BD	8\$:	RET		:	0855

; Routine Size: 190 bytes, Routine Base: _EDT\$CODE + 0000

; 286 0856 1
; 287 0857 1 !<BLF/PAGE>

0132 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

